

Unidade VIII:

Balanceamento de Árvores Binárias

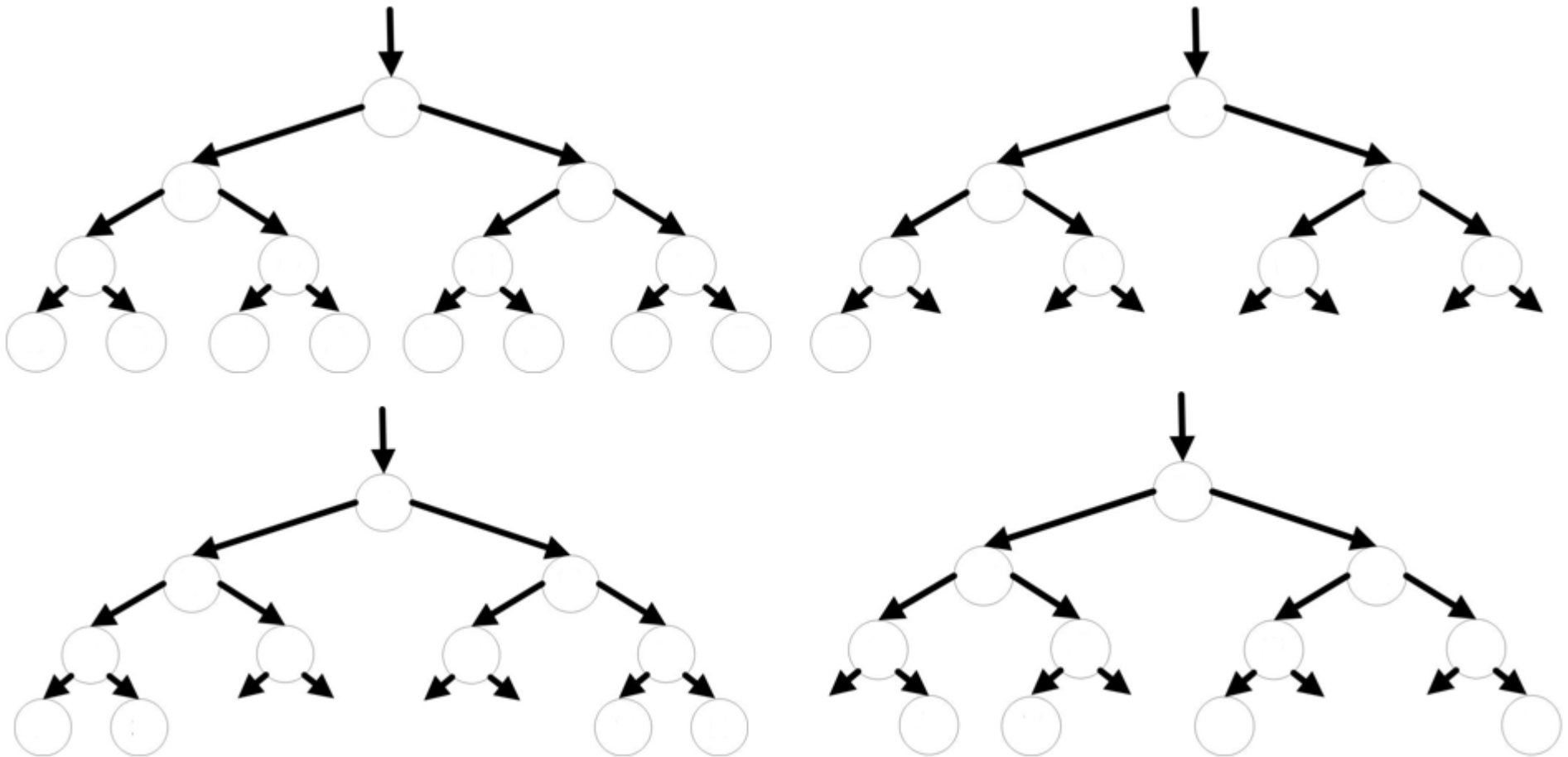


PUC Minas

Adaptação dos slides elaborados pelo Instituto de Ciências Exatas e
Informática - Departamento de Ciência da Computação

Árvore Balanceada

- Árvore em que para **TODOS** os nós, a diferença entre a altura de suas árvores da esquerda e da direita sempre será **0** ou **± 1** como, por exemplo:



Balanceamento de Árvores

- Qual é a vantagem de uma árvore balanceada?
 - Resposta: eficiência em termos de pesquisa, inserção e remoção
- Inicialmente, todas as árvores são balanceadas e elas podem desbalancear após as operações de inserção e remoção

Árvore Balanceada

- Minimiza o tempo médio de pesquisa:
 - Assumindo distribuição uniforme das chaves
- Problema: manter a árvore balanceada após cada inserção/remoção é muito caro.

Árvore pseudo-balanceada

- A solução é criar estruturas que mantenham um certo balanceamento mas que não exijam o balanceamento completo.
 - *Trade-off* entre custo de pesquisa e manutenção
- Exemplos:
 - Árvore 2-3-4
 - Árvores Vermelha e Preta
 - Árvore SBB
 - **Árvore AVL**


Balanceamento de Árvores – Operações

- As árvores desbalanceadas para a **esquerda** devem ser rotacionadas para a **direita**.
- As árvores desbalanceadas para a **direita** devem ser rotacionadas para a **esquerda**.

Tipos de Rotação

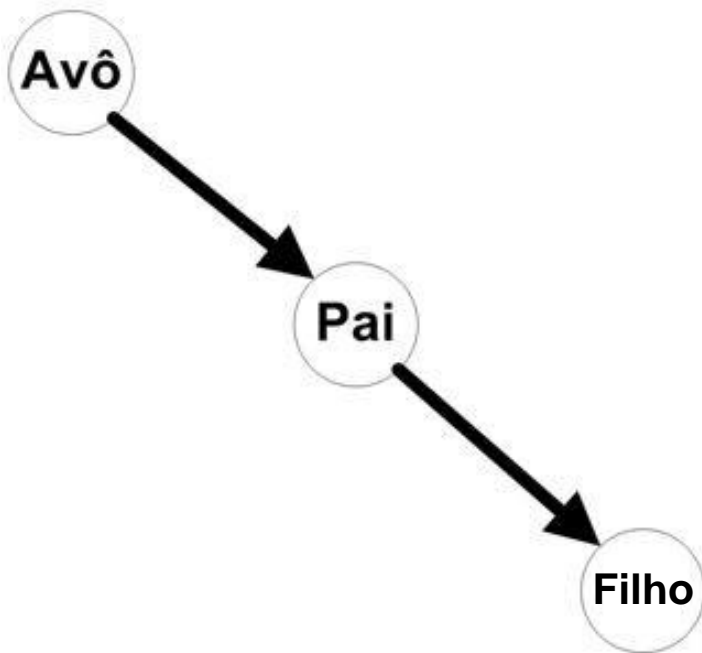
- Rotação simples à esquerda
- Rotação simples à direita
- Rotação dupla direita - esquerda
- Rotação dupla esquerda - direita

Tipos de Rotação

- **Rotação simples à esquerda** 
- Rotação simples à direita
- Rotação dupla direita - esquerda
- Rotação dupla esquerda - direita

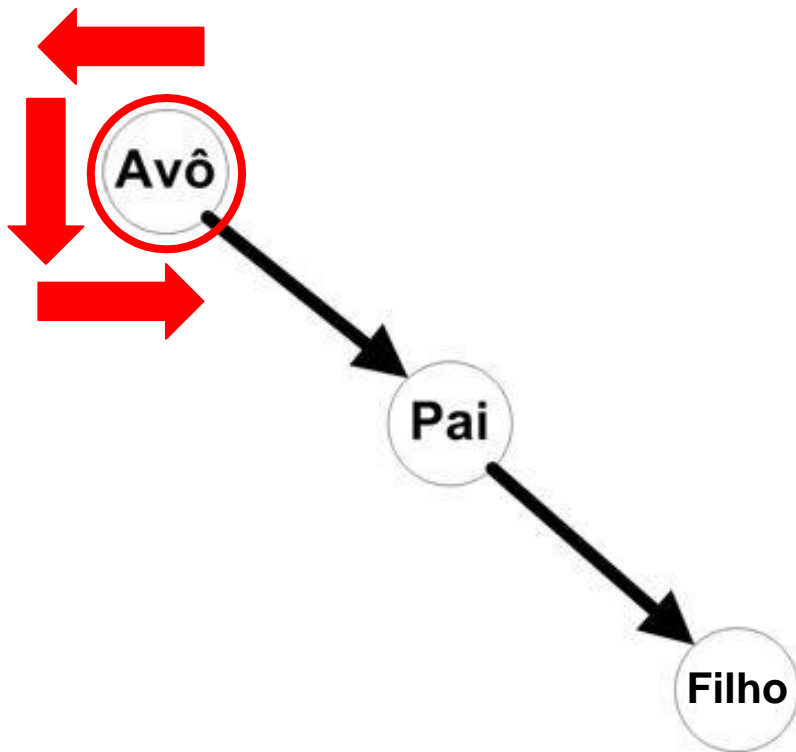
Rotação Simples à Esquerda

- Usada em subárvores em que o pai e o filho estão desbalanceados para a direita



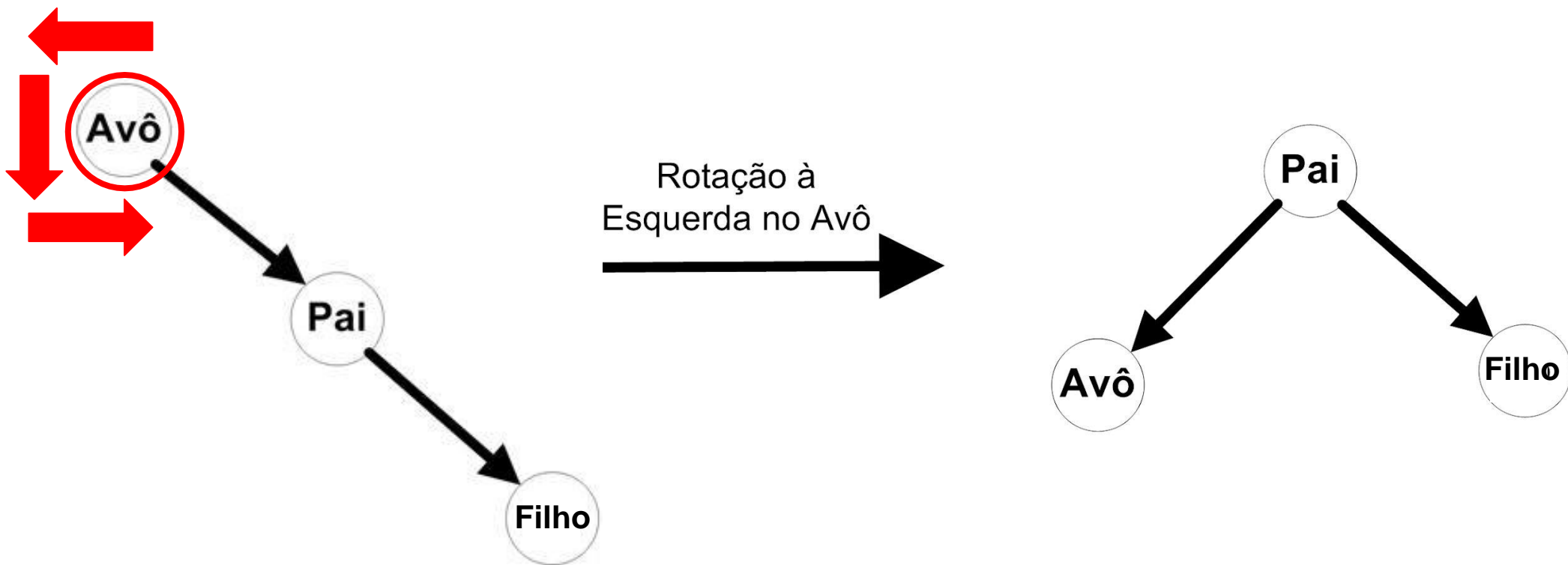
Rotação Simples à Esquerda

- Usada em subárvores em que o pai e o filho estão desbalanceados para a direita

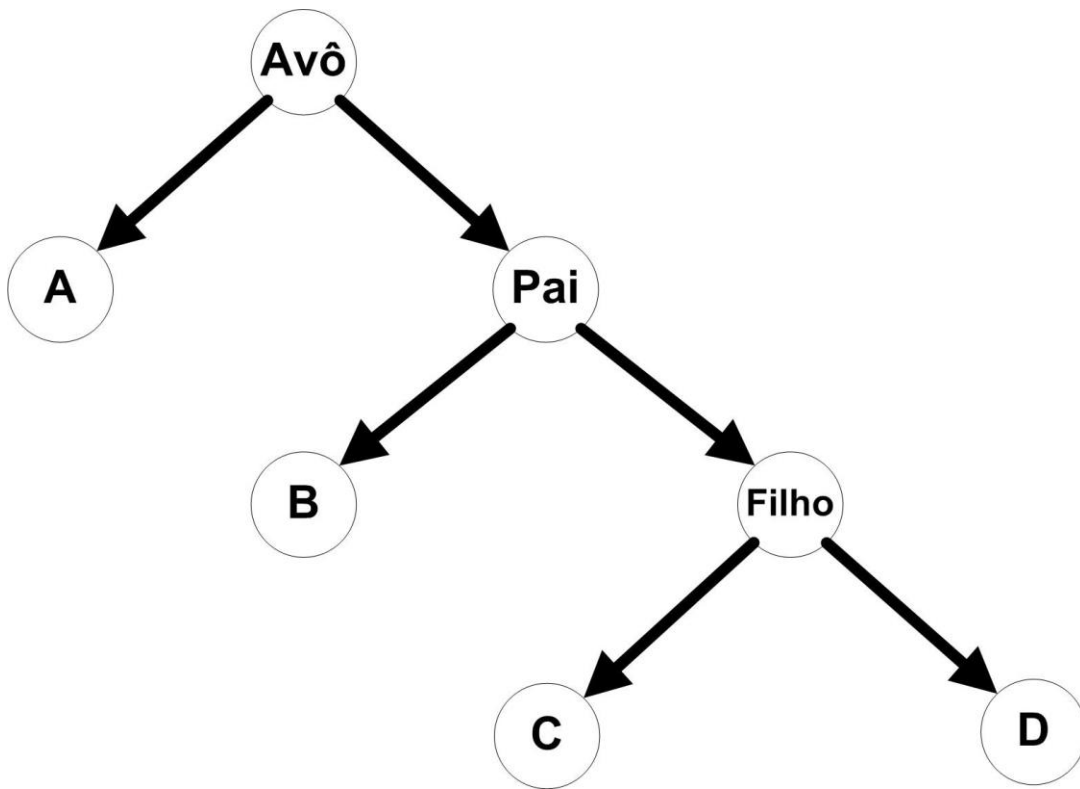


Rotação Simples à Esquerda

- Usada em subárvores em que o pai e o filho estão desbalanceados para a direita

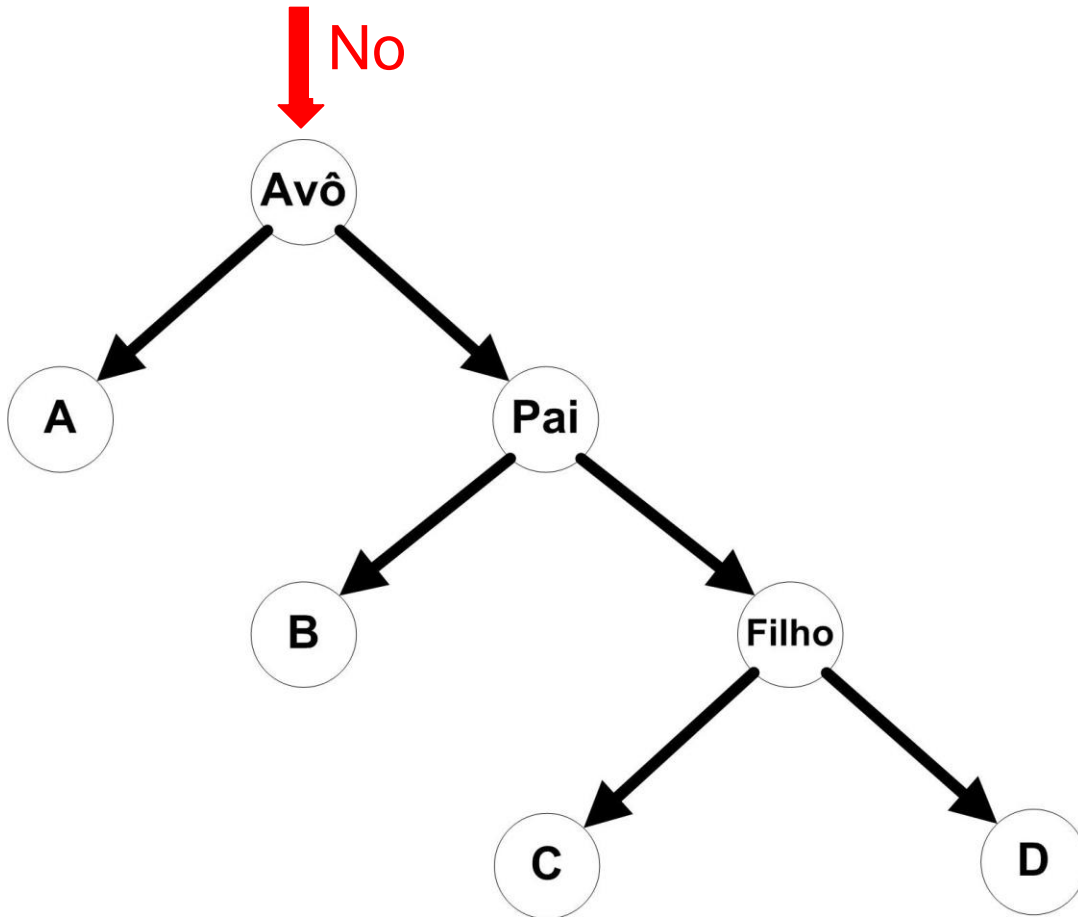


Implementação da Rotação à Esquerda



```
void metodo(){  
    ...  
    no = rotacionarEsq(no);  
    ...  
}  
No rotacionarEsq (No no) {  
    No noDir = no.Dir;  
    No noDirEsq = noDir.Esq;  
  
    noDir.Esq = no;  
    no.Dir = noDirEsq;  
  
    return noDir;  
}
```

Implementação da Rotação à Esquerda



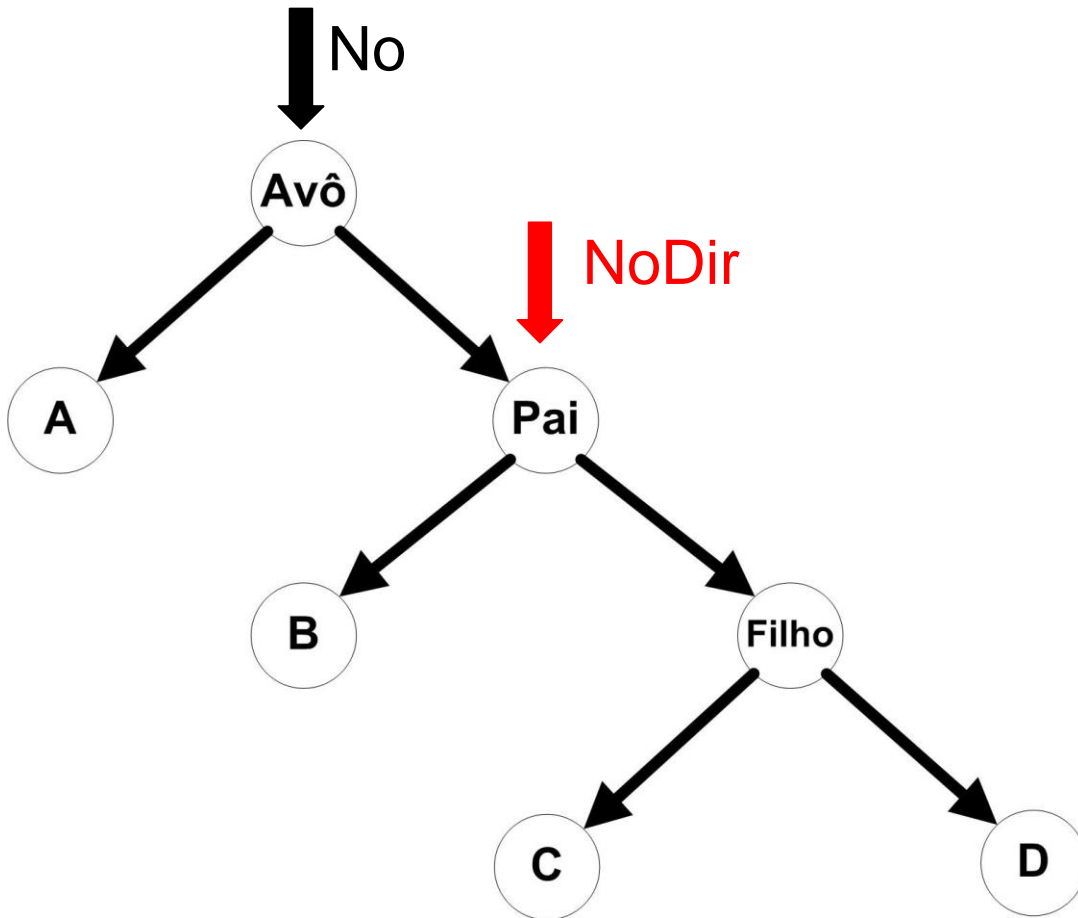
```

void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

    return noDir;
}
    
```

Implementação da Rotação à Esquerda



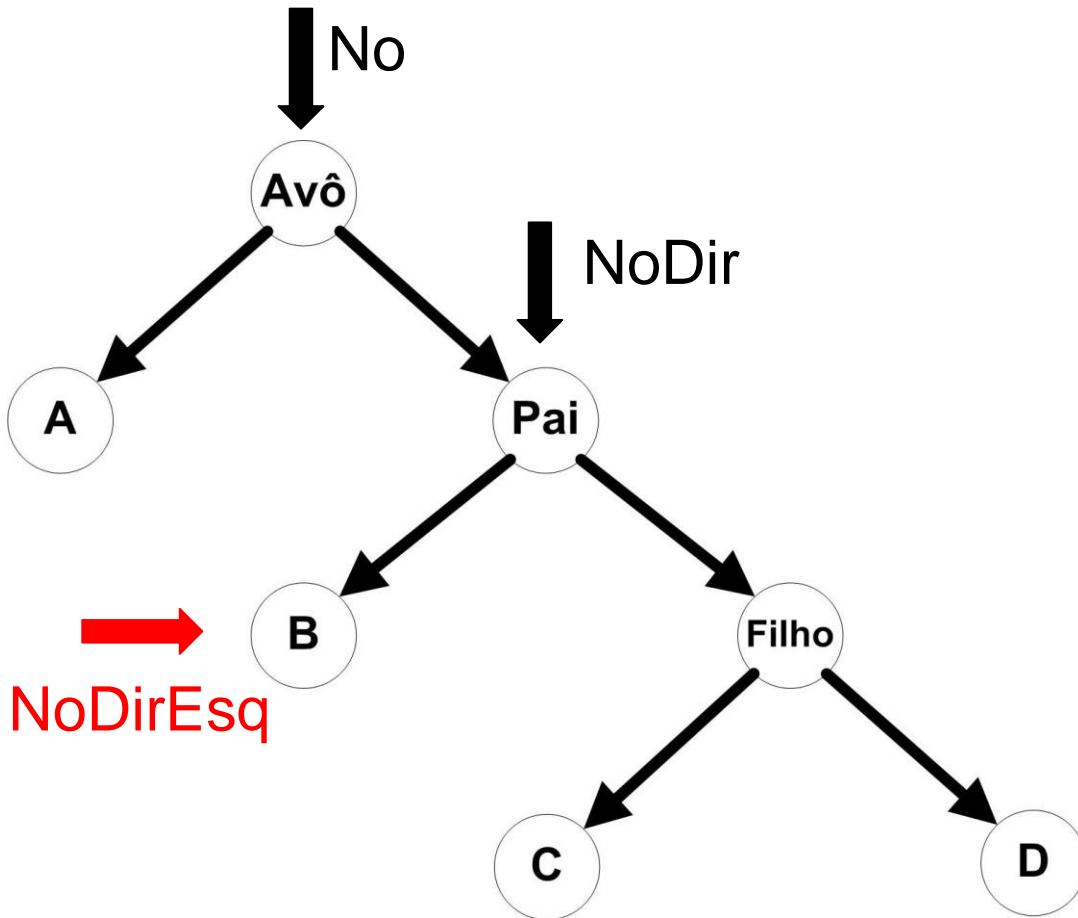
```

void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

    return noDir;
}
    
```

Implementação da Rotação à Esquerda



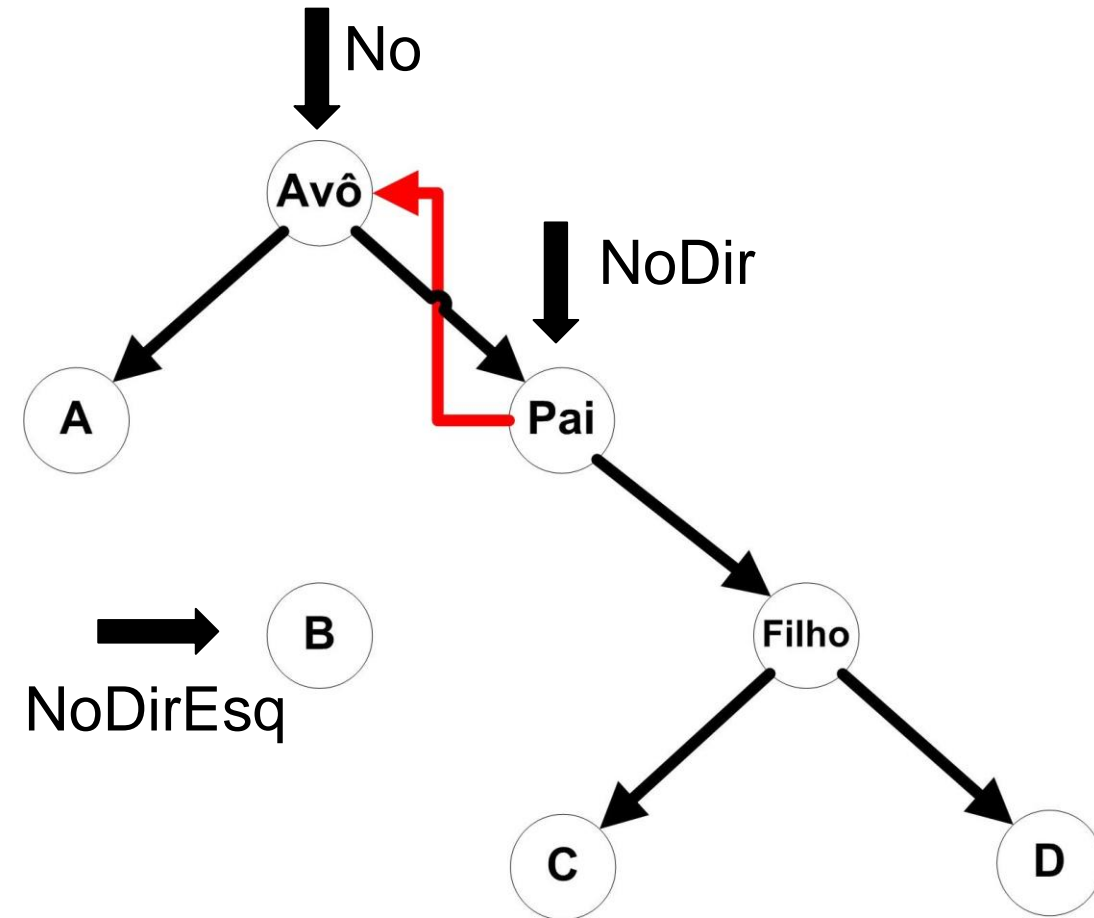
```

void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

    return noDir;
}
    
```

Implementação da Rotação à Esquerda

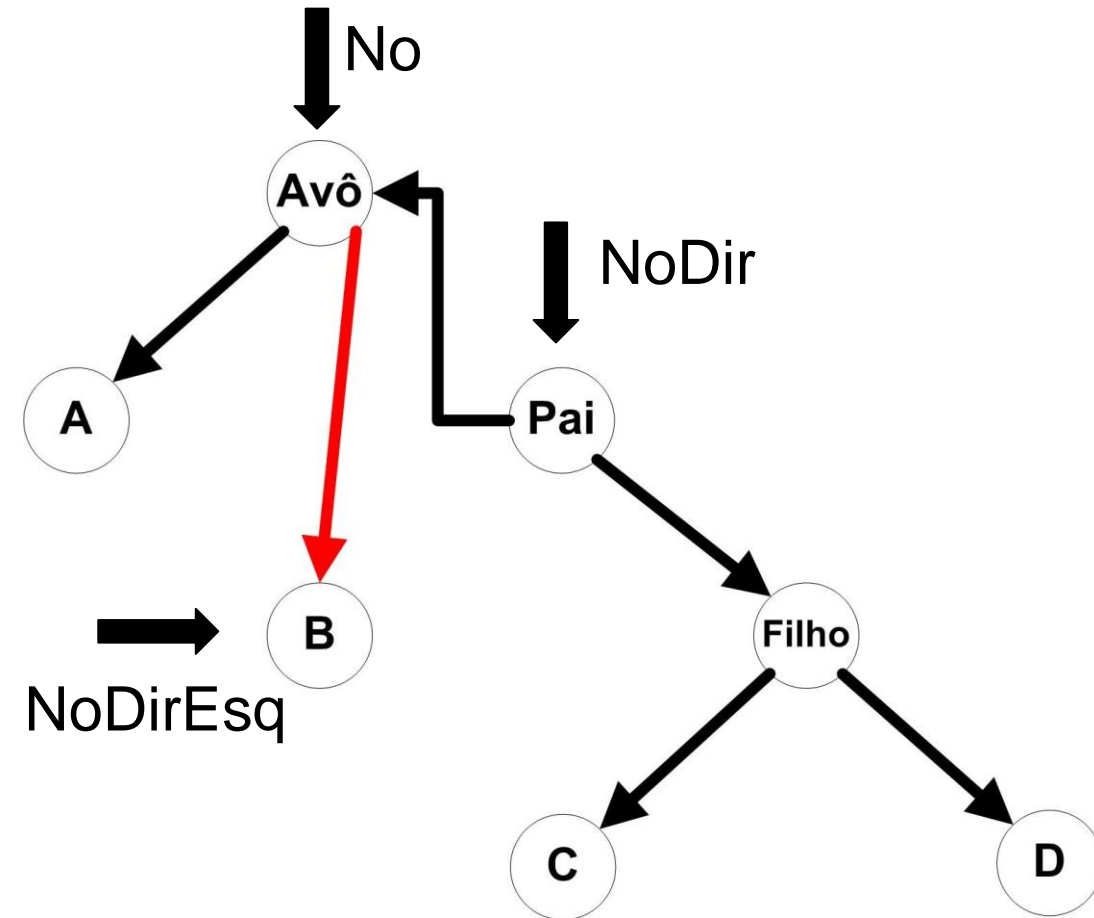


```
void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

    return noDir;
}
```


Implementação da Rotação à Esquerda

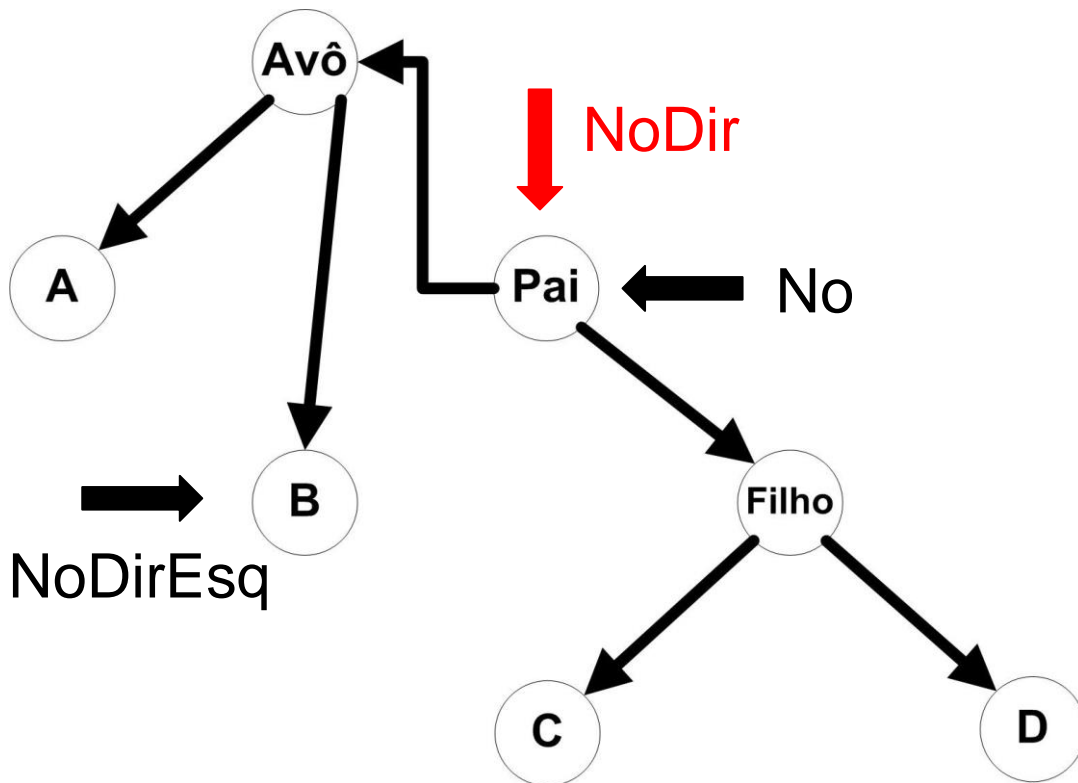


```
void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

    return noDir;
}
```

Implementação da Rotação à Esquerda



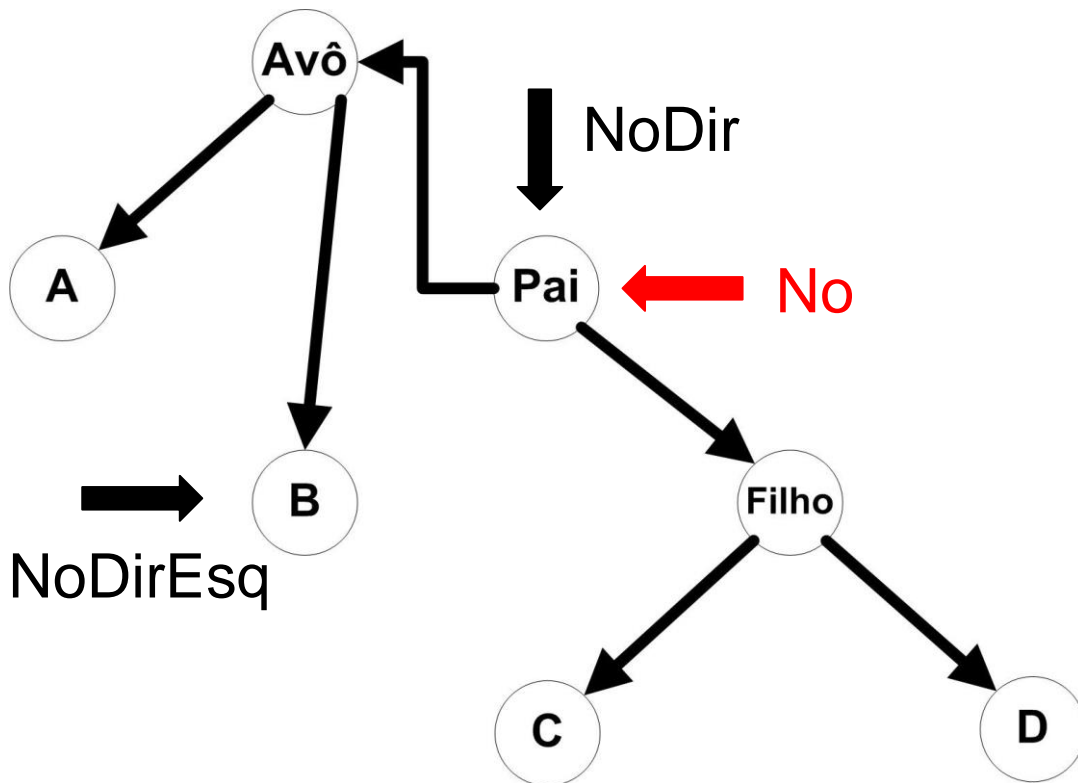
```

void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

    return noDir;
}
    
```

Implementação da Rotação à Esquerda



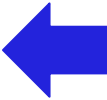
```

void metodo(){
    ...
    no = rotacionarEsq(no);
    ...
}
No rotacionarEsq (No no) {
    No noDir = no.Dir;
    No noDirEsq = noDir.Esq;

    noDir.Esq = no;
    no.Dir = noDirEsq;

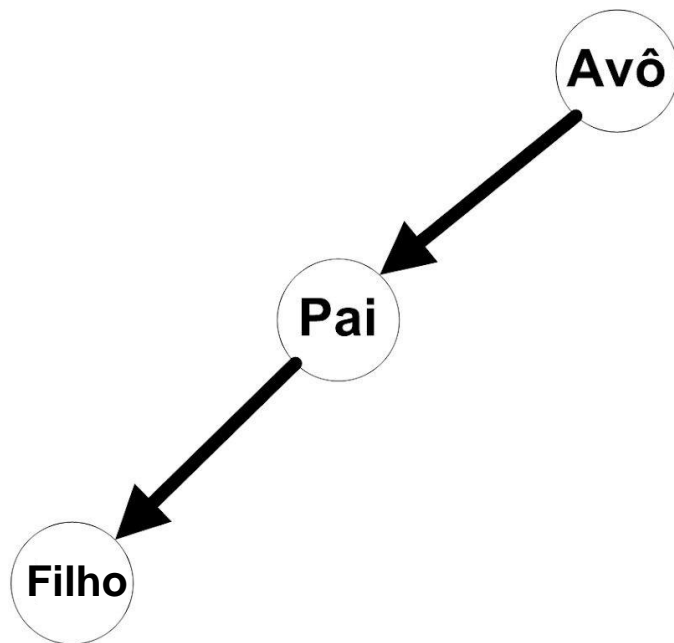
    return noDir;
}
    
```

Tipos de Rotação

- Rotação simples à esquerda
- **Rotação simples à direita** 
- Rotação dupla direita - esquerda
- Rotação dupla esquerda - direita

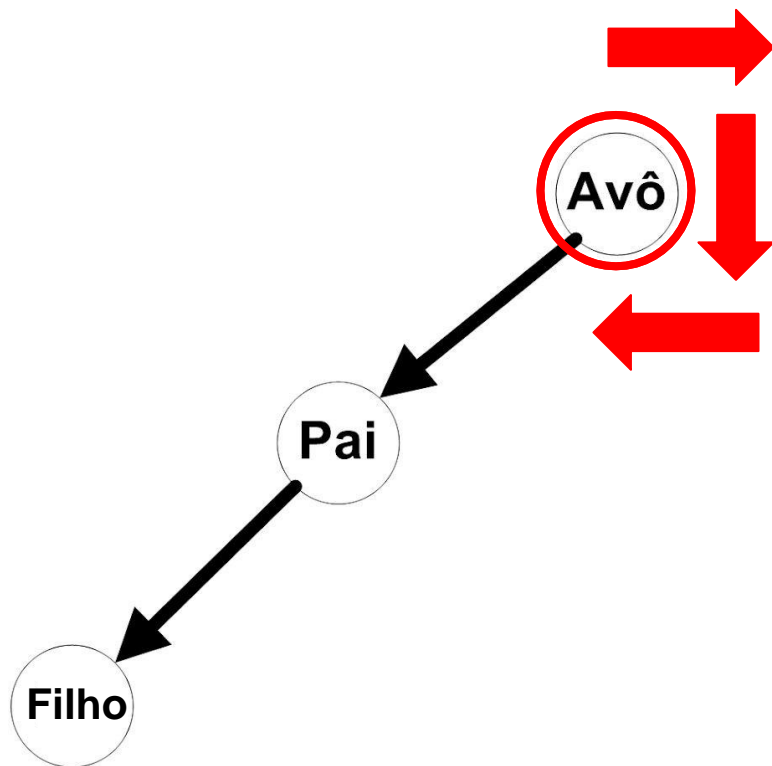
Rotação Simples à Direita

- Usada em subárvores em que o pai e o filho estão desbalanceados para a esquerda



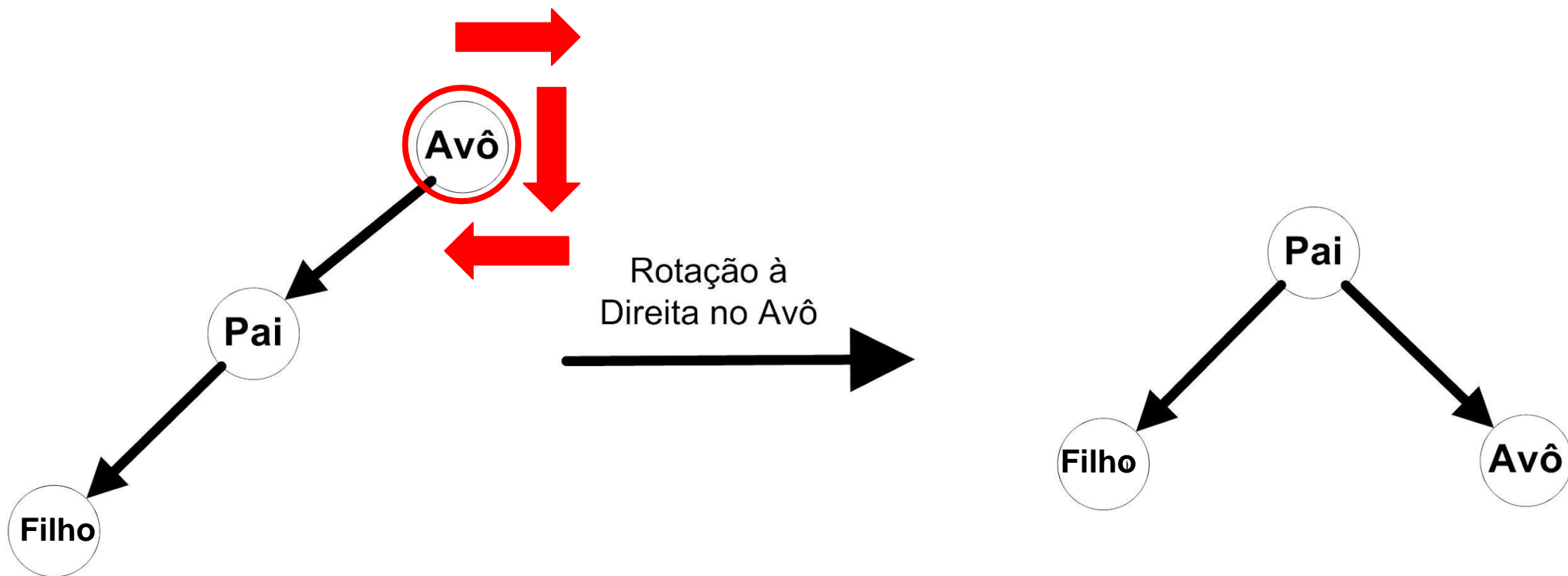
Rotação Simples à Direita

- Usada em subárvores em que o pai e o filho estão desbalanceados para a esquerda

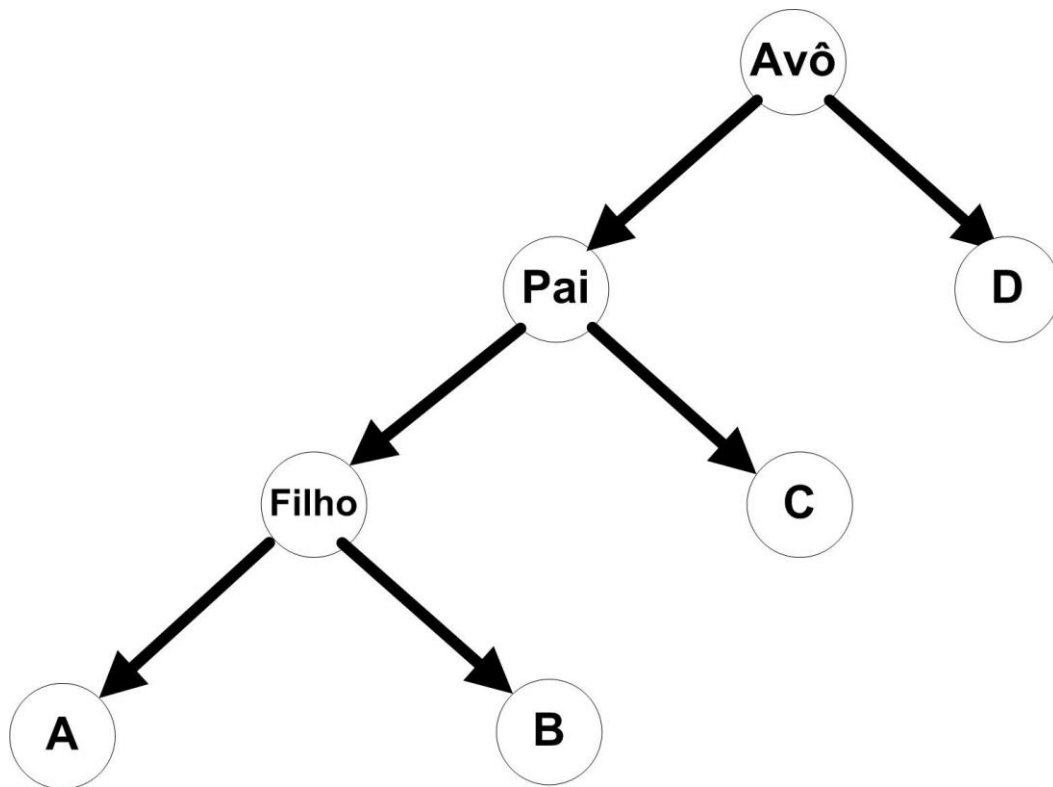


Rotação Simples à Direita

- Usada em subárvores em que o pai e o filho estão desbalanceados para a esquerda

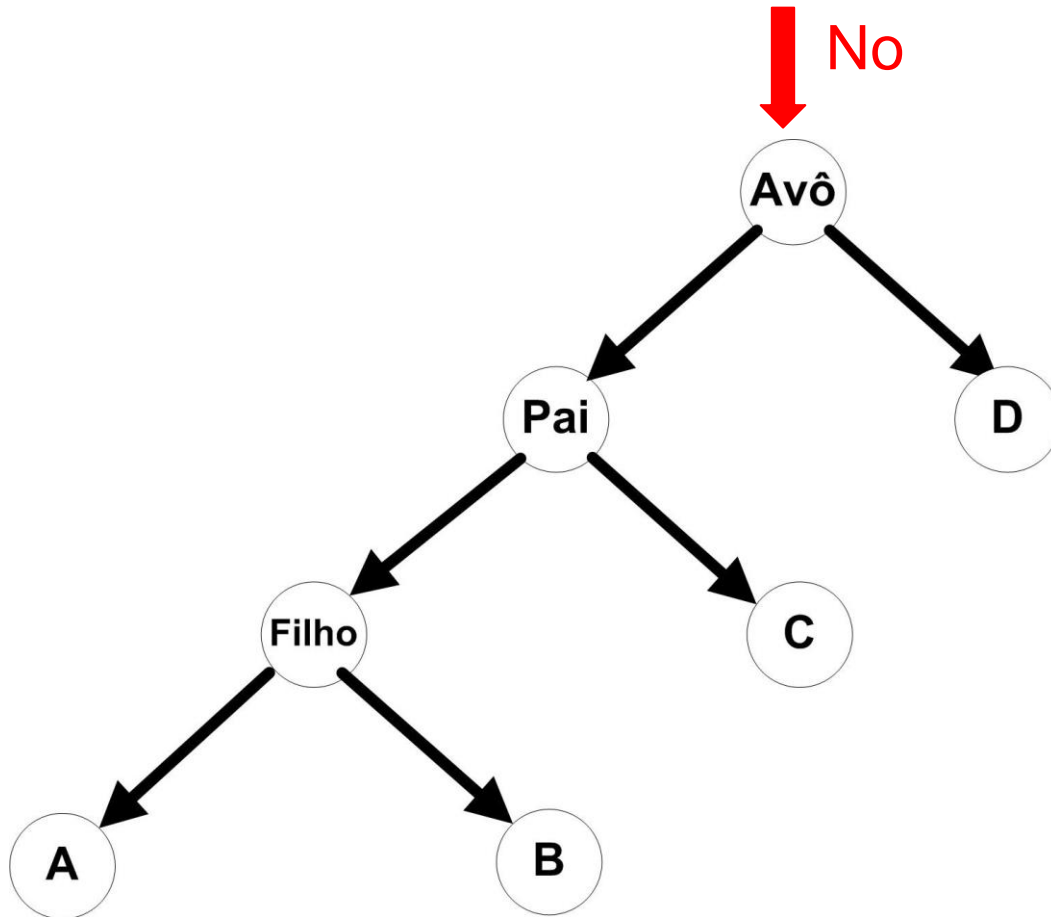


Implementação da Rotação à Direita



```
void metodo(){  
    ...  
    no = rotacionarDir(no);  
    ...  
}  
No rotacionarDir (No no) {  
    No noEsq = no.Esq;  
    No noEsqDir = noEsq.Dir;  
  
    noEsq.Dir = no;  
    no.Esq = noEsqDir;  
  
    return noEsq;  
}
```


Implementação da Rotação à Direita



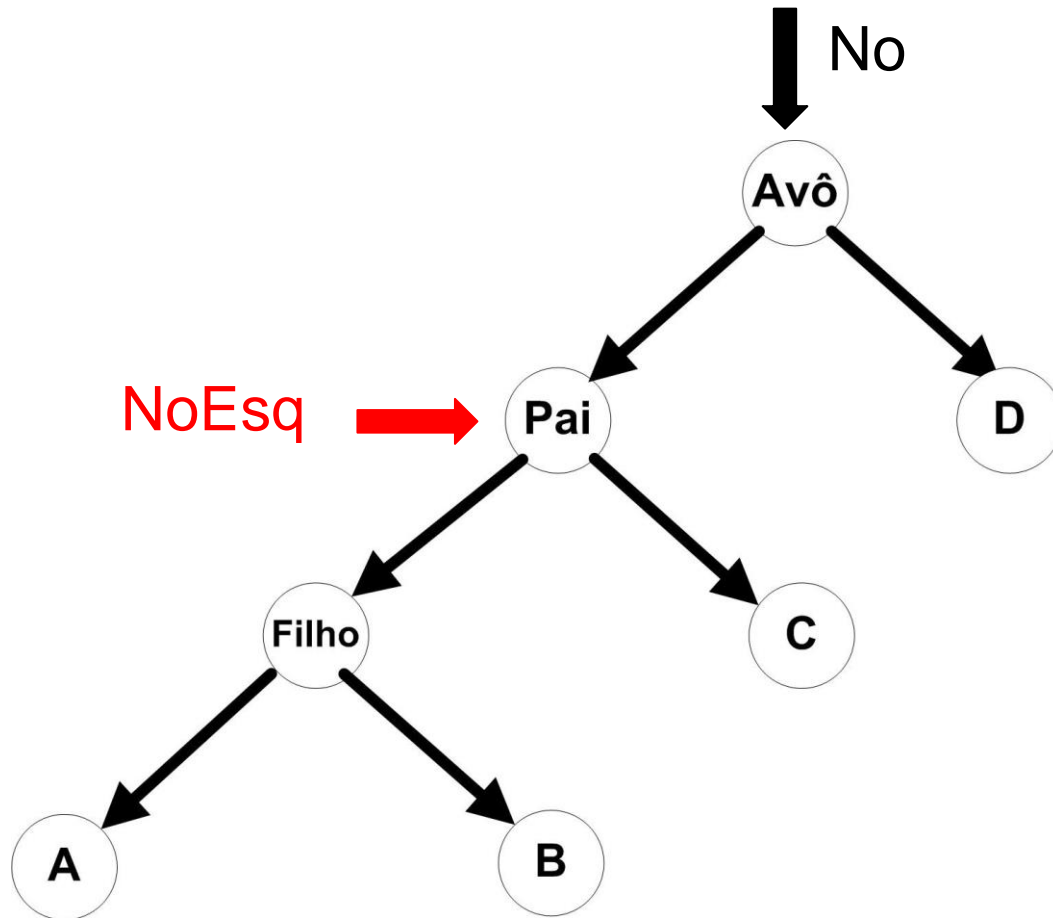
```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

    return noEsq;
}
    
```

Implementação da Rotação à Direita



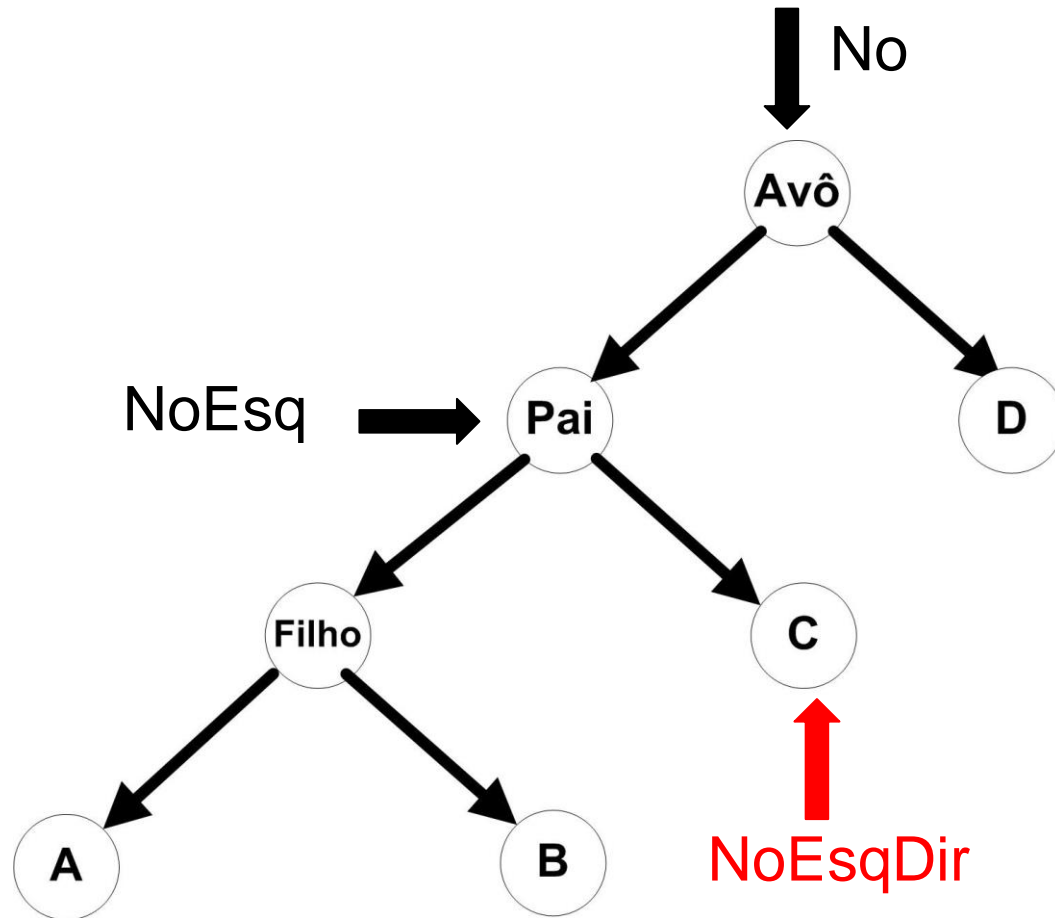
```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

    return noEsq;
}
    
```

Implementação da Rotação à Direita



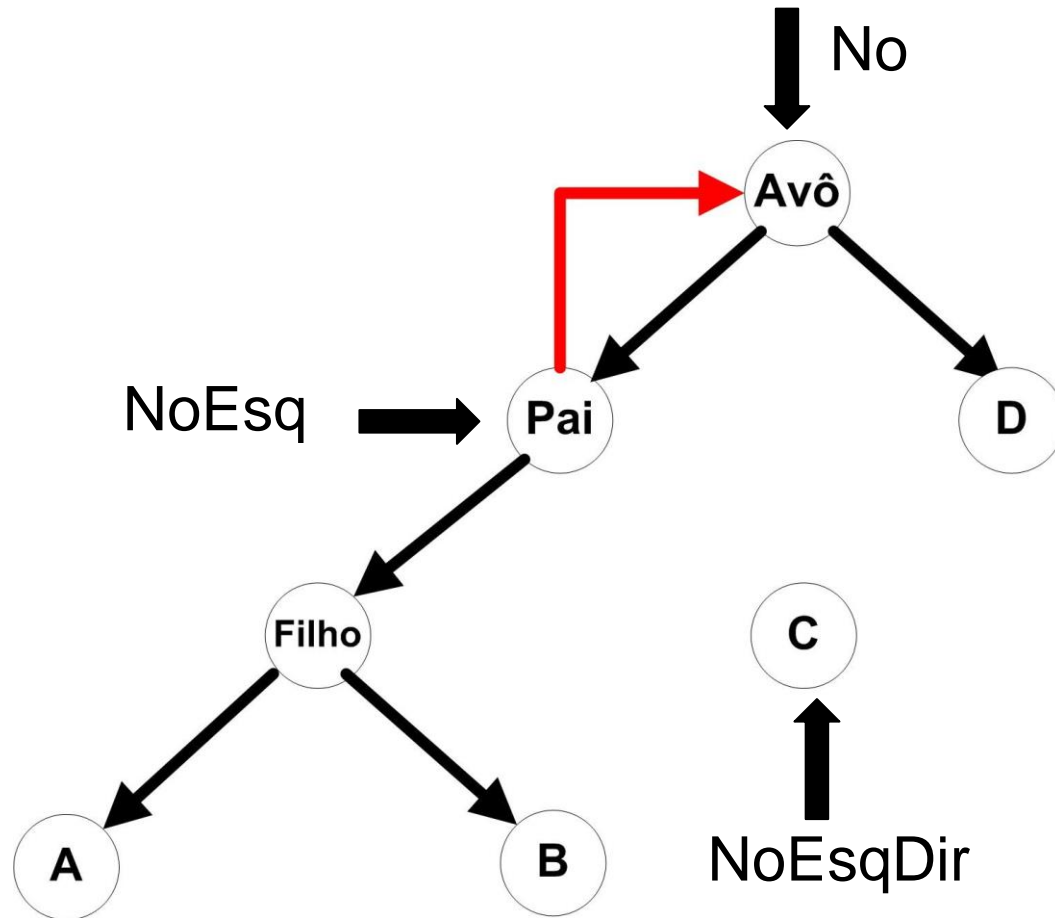
```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

    return noEsq;
}
    
```

Implementação da Rotação à Direita



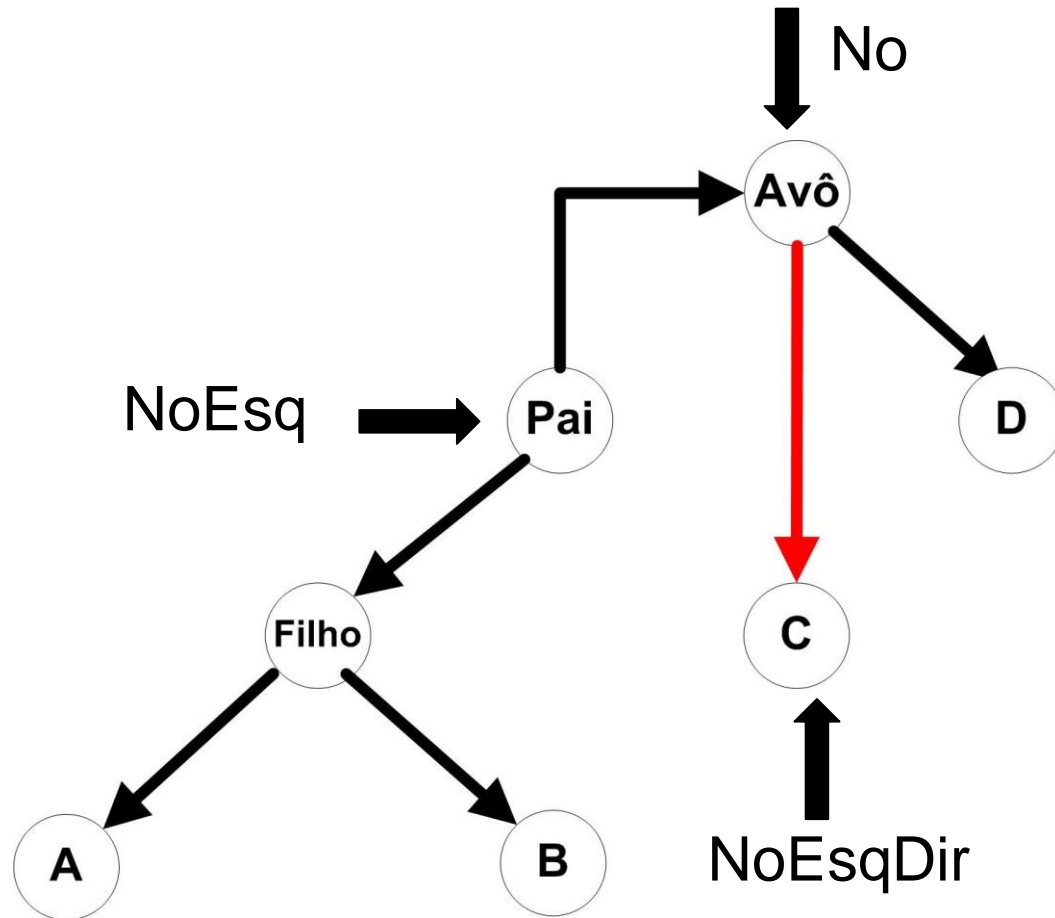
```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

    return noEsq;
}
    
```

Implementação da Rotação à Direita



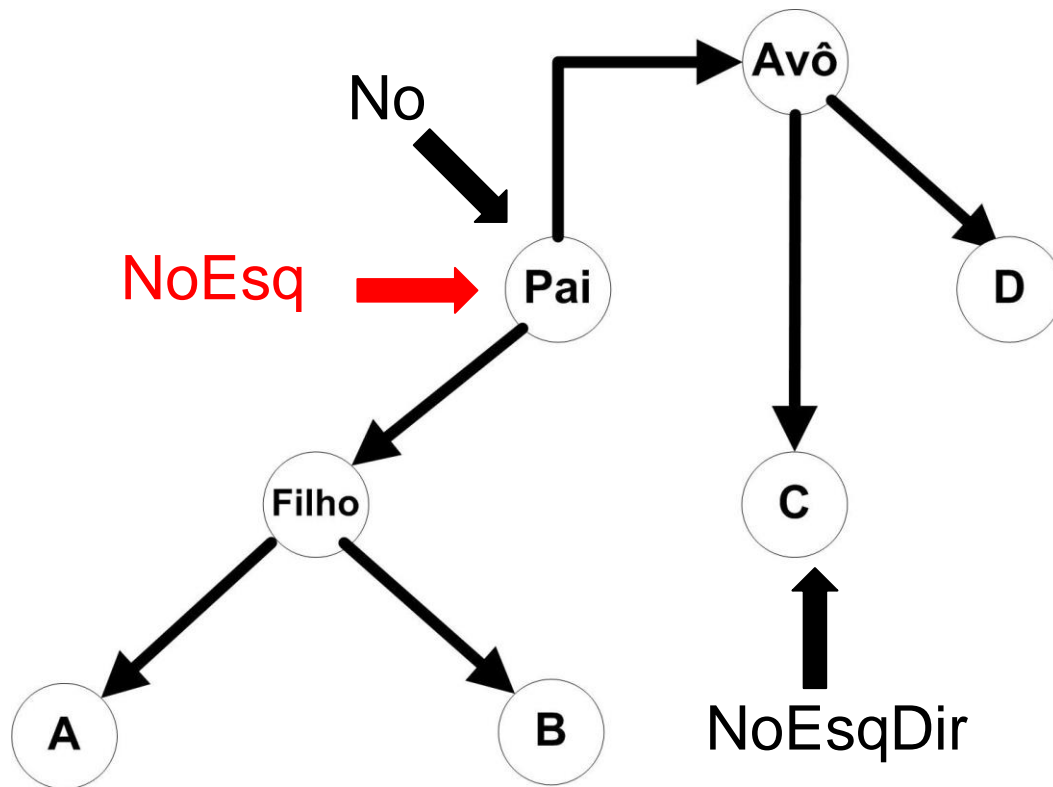
```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

    return noEsq;
}
    
```

Implementação da Rotação à Direita



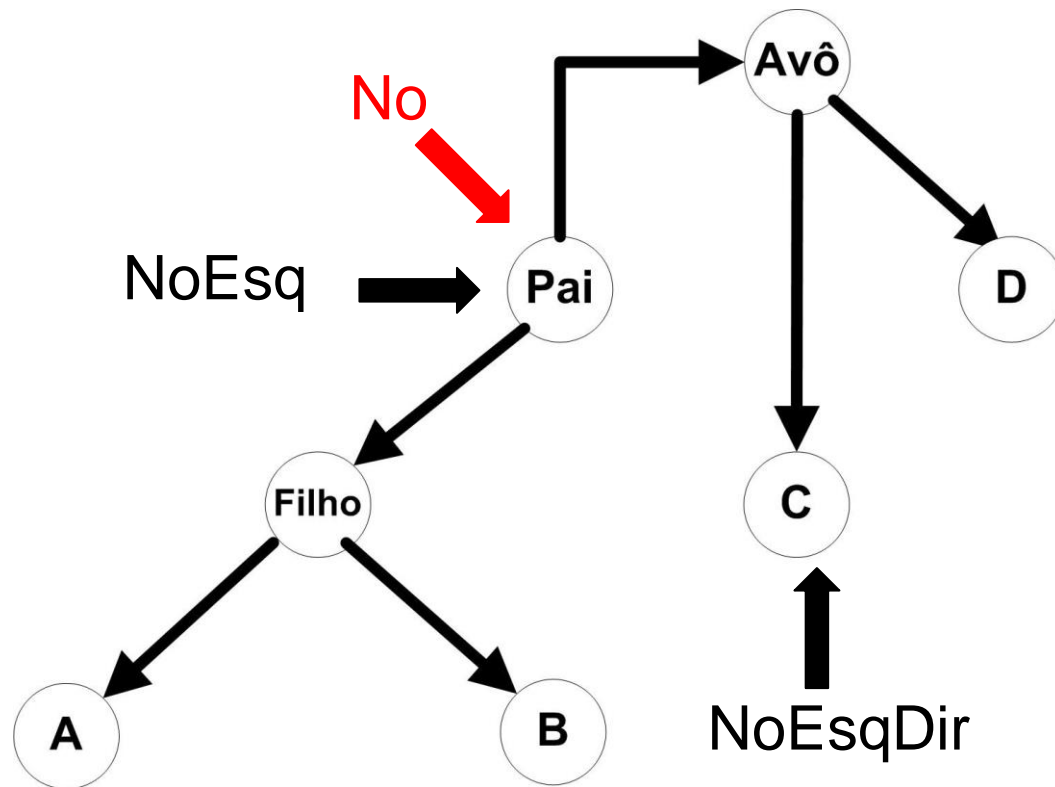
```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

    return noEsq;
}
    
```

Implementação da Rotação à Direita




```

void metodo(){
    ...
    no = rotacionarDir(no);
    ...
}
No rotacionarDir (No no) {
    No noEsq = no.Esq;
    No noEsqDir = noEsq.Dir;

    noEsq.Dir = no;
    no.Esq = noEsqDir;

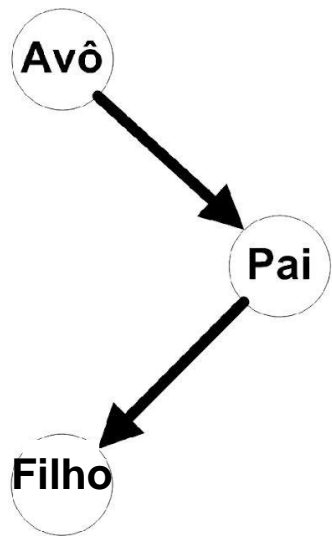
    return noEsq;
}
    
```

Tipos de Rotação

- Rotação simples à esquerda
- Rotação simples à direita
- **Rotação dupla direita - esquerda** 
- Rotação dupla esquerda - direita

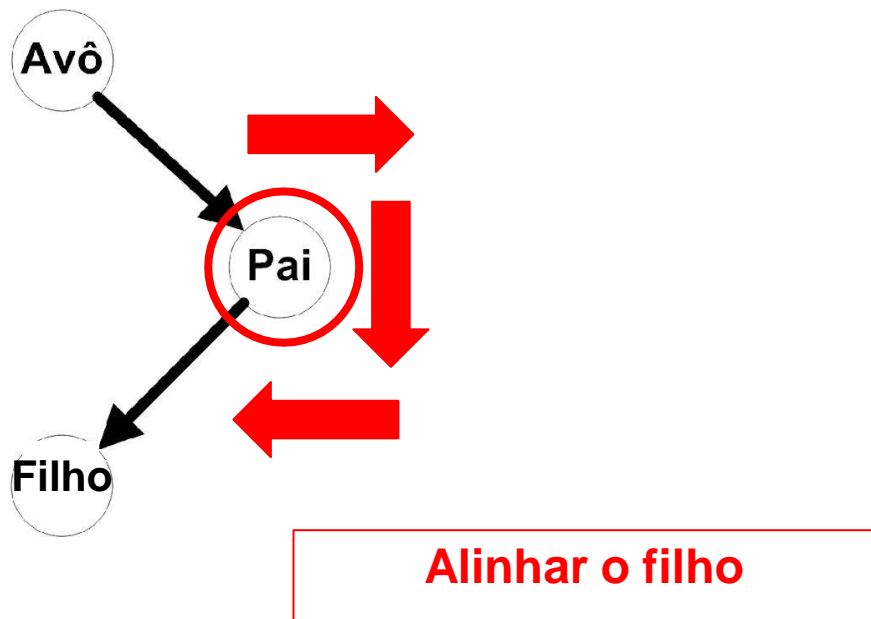
Rotação Dupla Direita - Esquerda

- Usada em subárvores em que um nó está desbalanceado para a direita e seu filho para a esquerda



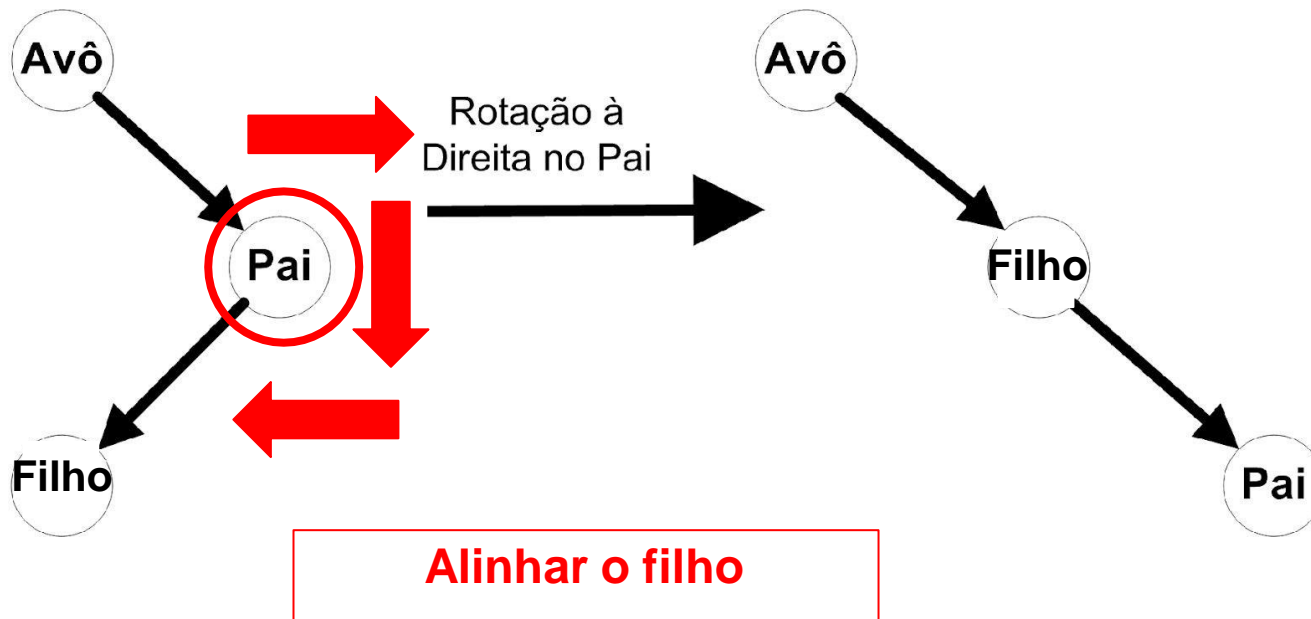
Rotação Dupla Direita - Esquerda

- Usada em subárvores em que um nó está desbalanceado para a direita e seu filho para a esquerda



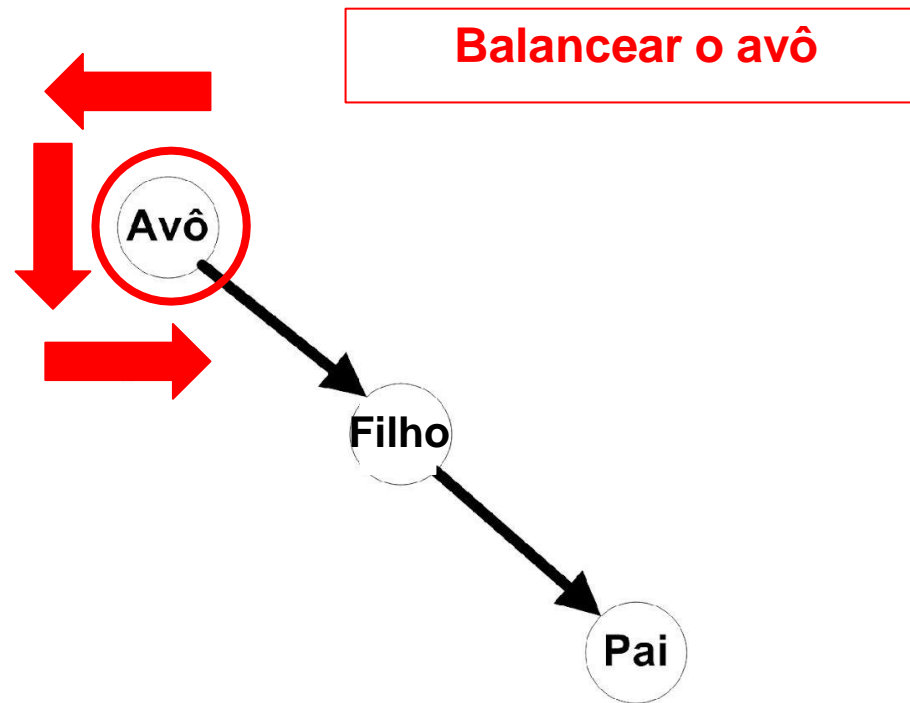
Rotação Dupla Direita - Esquerda

- Usada em subárvores em que um nó está desbalanceado para a direita e seu filho para a esquerda



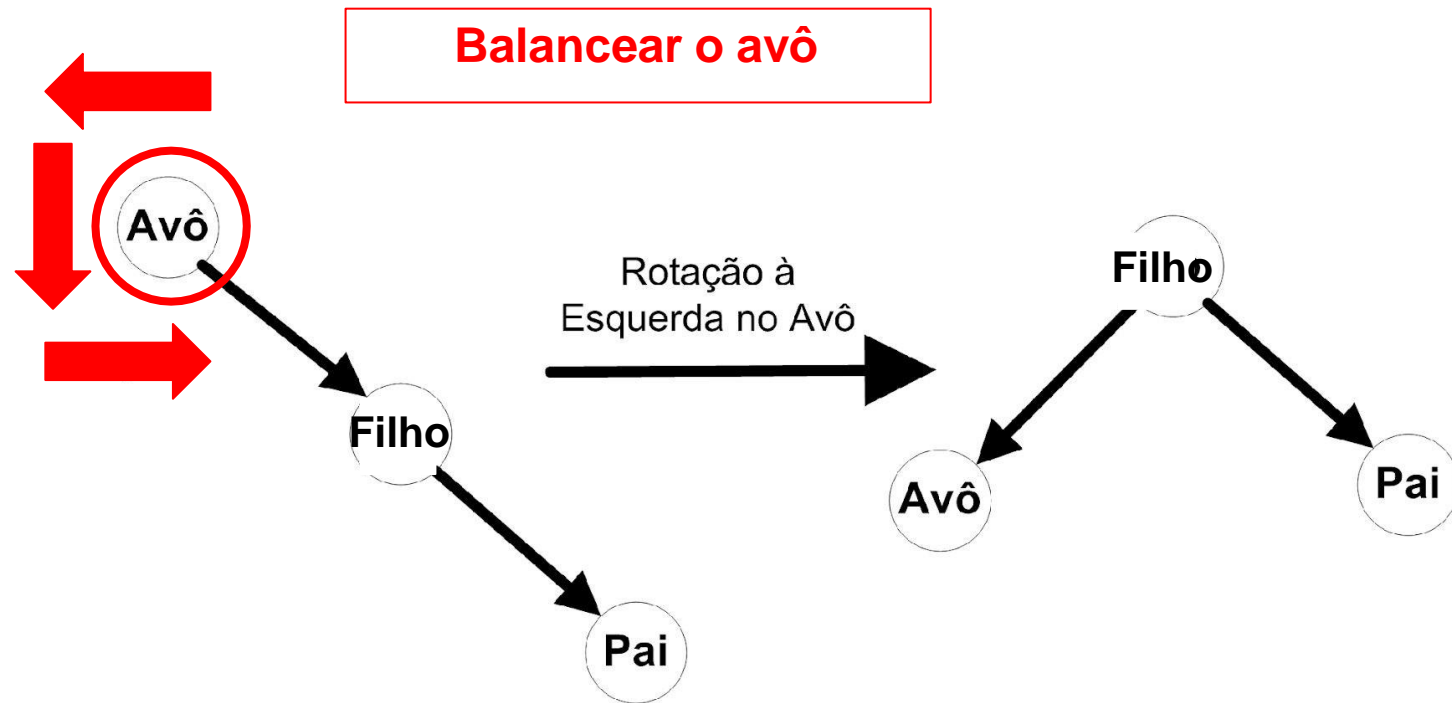
Rotação Dupla Direita - Esquerda

- Usada em subárvores em que um nó está desbalanceado para a direita e seu filho para a esquerda



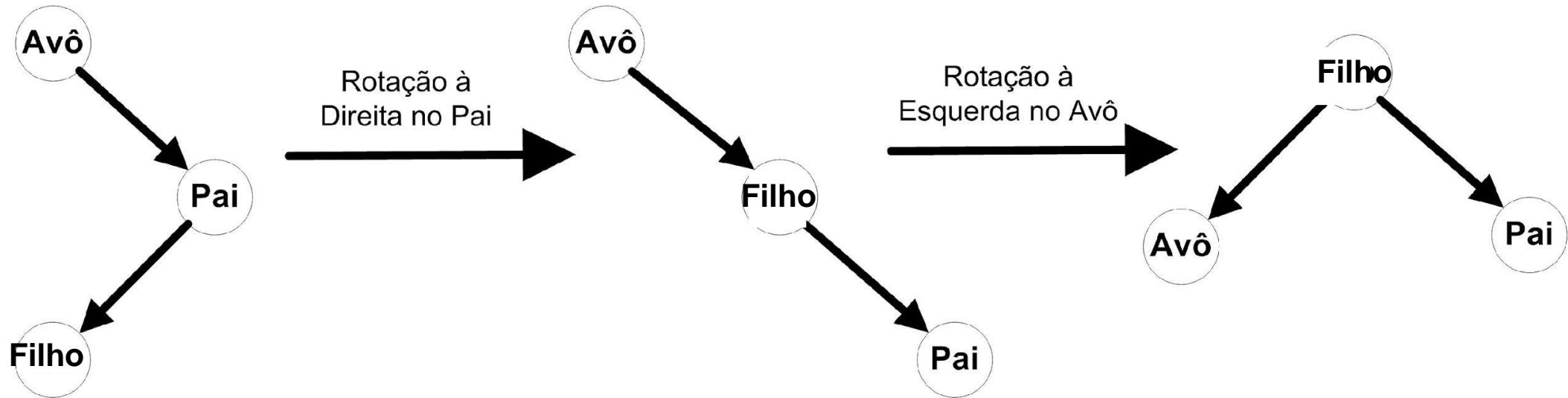
Rotação Dupla Direita - Esquerda

- Usada em subárvores em que um nó está desbalanceado para a direita e seu filho para a esquerda




Implementação da Rotação à Direita - Esquerda

```
No rotacionarDirEsq(No no) {  
    no.Dir = rotacionarDir(no.Dir);  
    return rotacionarEsq(no);  
}
```

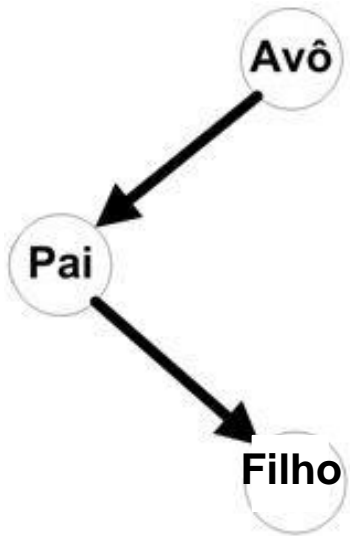


Tipos de Rotação

- Rotação simples à esquerda
- Rotação simples à direita
- Rotação dupla direita - esquerda
- **Rotação dupla esquerda - direita** 

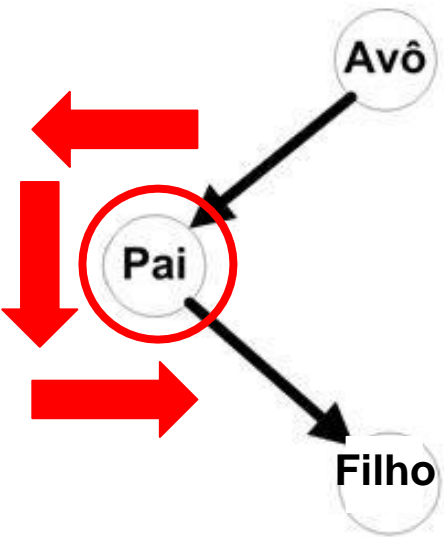
Rotação Dupla Esquerda - Direita

- Usada em subárvores em que um nó está desbalanceado para a esquerda e seu filho para a direita



Rotação Dupla Esquerda - Direita

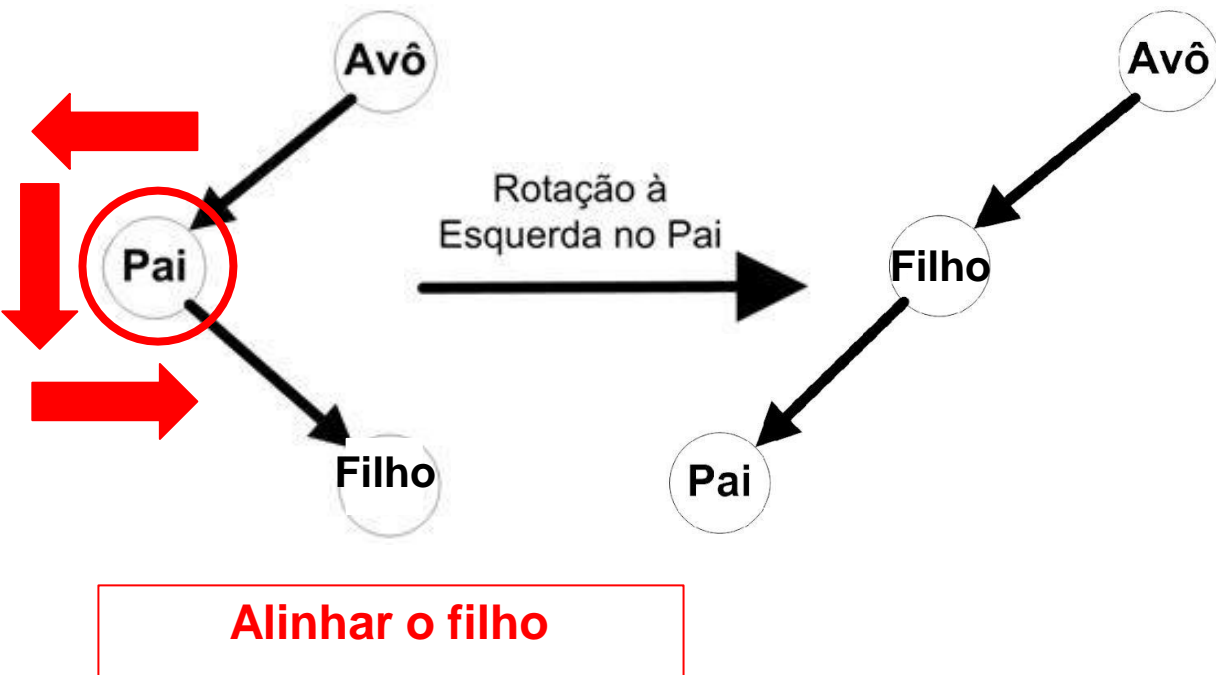
- Usada em subárvores em que um nó está desbalanceado para a esquerda e seu filho para a direita



Alinhar o filho

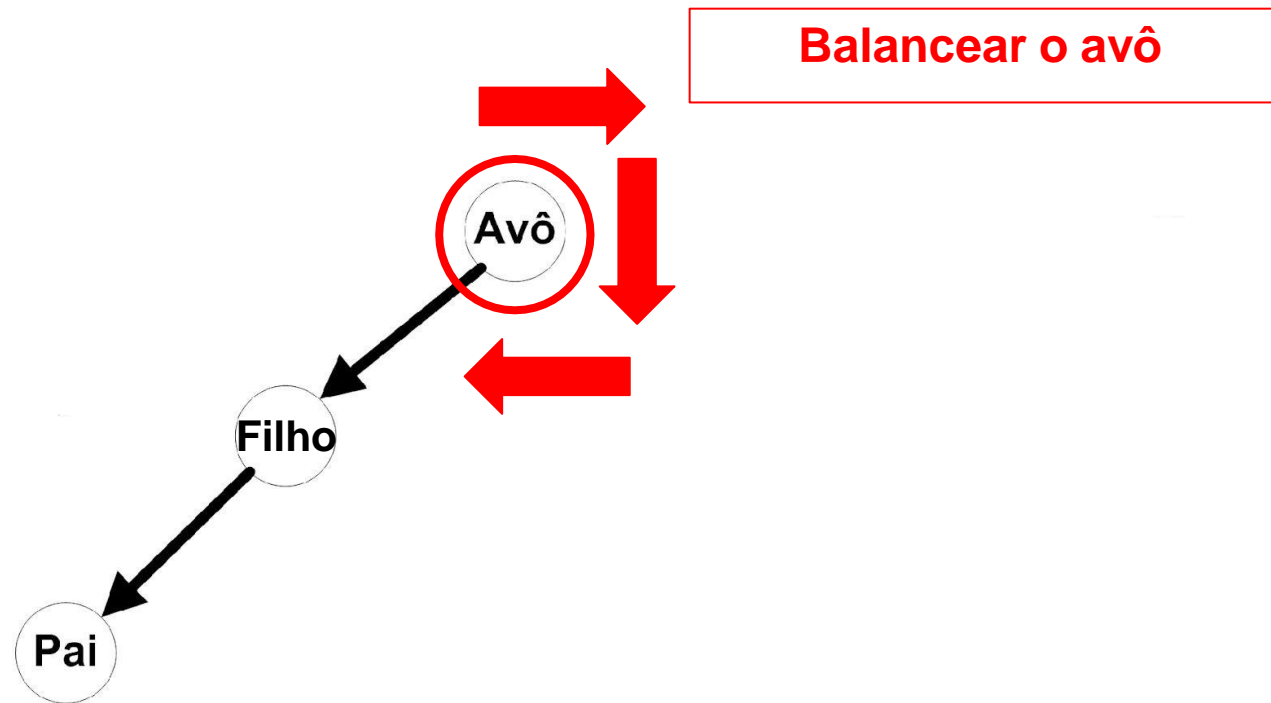
Rotação Dupla Esquerda - Direita

- Usada em subárvores em que um nó está desbalanceado para a esquerda e seu filho para a direita



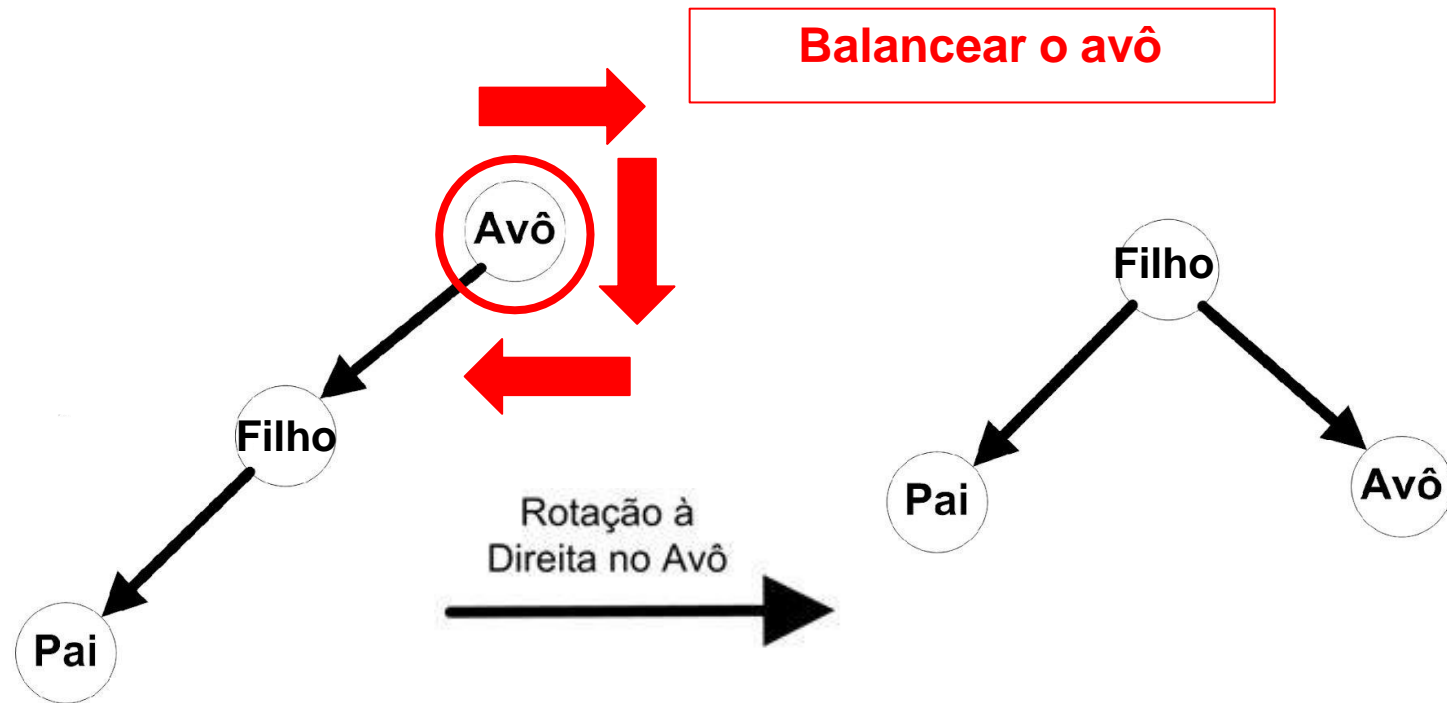
Rotação Dupla Esquerda - Direita

- Usada em subárvores em que um nó está desbalanceado para a esquerda e seu filho para a direita



Rotação Dupla Esquerda - Direita

- Usada em subárvores em que um nó está desbalanceado para a esquerda e seu filho para a direita



Implementação da Rotação à Esquerda - Direita

```
No rotacionarEsqDir(No no) {  
    no.Esq = rotacionarEsq(no.Esq);  
    return rotacionarDir(no);  
}
```

